

Universidade da Beira Interior

Departamento de Informática



Nº 172 - 2021: *As minhas dúvidas*

Elaborado por:

Rodrigo Dos Santos Gil Saraiva

Orientador:

Professora Doutora Maria Paula Prata de Sousa

23 de abril de 2021

Agradecimentos

A realização deste projeto, não seria possível sem a ajuda e a motivação de várias pessoas, deste modo, queria agradecer à minha orientadora Professora Maria Paula Prata de Sousa, pelo apoio prestado ao longo do semestre e sua constante disponibilidade.

Gostaria de agradecer também à minha família pela força, encorajamento e benevolência proporcionada no decorrer desta caminhada.

Aos meus amigos e colegas, pelo apoio prestado e a todos os que sempre acreditaram em mim.

Conteúdo

Conteúdo	iii
Lista de Figuras	v
Lista de Tabelas	vii
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação	1
1.3 Objectivos	2
1.4 Organização do Documento	2
2 Estado da Arte e Contextualização Tecnológica	3
2.1 Introdução	3
2.2 Contextualização Tecnológica	3
2.2.1 Bibliotecas Usadas	5
2.2.2 Ferramentas Usadas	5
2.3 Estado de Arte	5
2.3.1 Stackoverflow[1]	6
2.3.2 Experts-Exchange	6
2.4 Conclusões	7
3 Engenharia de Software	9
3.1 Introdução	9
3.2 Análise de Requisitos	9
3.3 Casos de Uso	12
3.4 Base de Dados	17
3.5 Conclusões	18
4 Implementação e Testes	19
4.1 Introdução	19
4.2 Servidor	19
4.3 Aplicação	23
4.3.1 Página Inicial	24

4.3.2	Menu	25
4.3.3	Registrar	26
4.3.4	Confirmar registo	27
4.3.5	Login	28
4.3.6	Repor password	28
4.3.7	Alteração de password	29
4.3.8	Apoio ao aluno	30
4.3.9	Criar cadeira e curso	33
4.3.10	Dúvidas	34
4.3.11	Visualizar Dúvida	37
4.3.12	Fórum	38
4.4	Segurança	42
4.5	Testes	44
4.6	Conclusões	44
5	Conclusões e Trabalho Futuro	45
5.1	Conclusões Principais	45
5.2	Trabalho Futuro	45
Bibliografia		47

Listas de Figuras

2.1	Arquitetura ASP NET Core[2].	4
2.2	Página do Stackoverflow	6
2.3	Case sensitive Win31 HTML Editor	7
3.1	Caso de uso login.	13
3.2	Caso de uso registo.	14
3.3	Caso de uso menu.	15
3.4	Caso de uso visualizar dúvida.	16
3.5	Base de dados.	17
4.1	Demonstração das páginas responsivas.	24
4.2	Página Inicial.	25
4.3	Página Inicial.	26
4.4	Página do registo.	27
4.5	Página informativa do registo.	27
4.6	Página do login.	28
4.7	Página que permite repor a password.	29
4.8	Página que permite alterar a password.	30
4.9	Página obtida após alteração da password.	30
4.10	Tab da página apoio ao aluno que correspondente à área cadeira. .	31
4.11	Tab da página apoio ao aluno que permite parar de seguir uma cadeira.	32
4.12	Tab da página apoio ao aluno que permite seguir uma cadeira. .	33
4.13	Página que permite criar um curso.	34
4.14	Página que permite criar uma cadeira.	34
4.15	Página que lista as dúvidas existentes.	35
4.16	Criar dúvida.	36
4.17	Visualizar dúvida.	38
4.18	Resposta a uma dúvida.	38
4.19	Visualização dos fóruns.	39
4.20	Criação de um fórum.	39
4.21	Visualização dos posts de um fórum.	40
4.22	Criação de um post.	41
4.23	Editar fórum.	41

Listas de Tabelas

3.1 Requisitos Funcionais	11
3.2 Requisitos Não Funcionais	11

Listas de Excertos de Código

4.1	Criação da chave privada.	20
4.2	Criação de um certificado auto-assinado.	20
4.3	Criação de um ficheiro pfx.	20
4.4	Comando para confiar no certificado.	20
4.5	Código que permite configurar o kestrel.	21
4.6	Configuração do HttpServer.	21
4.7	Comando que guarda a string "HttpServer:Endpoints:Https>Password".	21
4.8	Configuração do endereço HTTPS da aplicação.	22
4.9	Comando que permite o armazenamento da senha do certificado de forma segura.	22
4.10	Implementação do email JobScheduler.	23
4.11	Anotação AllowAnonymous.	42
4.12	Anotação Authorize.	42
4.13	Anotação Authorize(Roles: "aluno").	42
4.14	Anotação Authorize(Roles: "professor").	43
4.15	Anotação Authorize(Roles: "admin").	43
4.16	Comando que esconde a string "ConnectionStrings:DefaultConnection".	43
4.17	Comando que esconde a password do email.	44

Acrónimos

BD	Base Dados
CoreCLR	Common Language Runtime
IHC	Interação Humana com o Computador
MVC	Model View Controller
SD	Sistemas Distribuídos
SI	Segurança Informática
UBI	Universidade da Beira Interior
UC	Unidade Curricular

Capítulo

1

Introdução

Neste capítulo é feito um enquadramento do projeto, com a posterior apresentação do motivo da escolha deste tema, finalizando com a apresentação da estrutura deste documento.

1.1 Enquadramento

Este projeto foi desenvolvido no âmbito da Unidade Curricular (UC) Projeto do curso de Engenharia Informática da Universidade da Beira Interior (UBI) abrangendo conhecimentos de algumas áreas estudadas até ao momento, essas mesmas são Sistemas Distribuídos (SD), Base Dados (BD), Interação Humana com o Computador (IHC), Segurança Informática (SI) entre outras.

1.2 Motivação

O principal motivo para a escolha deste projeto está relacionado com o interesse pessoal de aprofundar conhecimentos na área de SD. Com isto em mente decidi abordar a professora Maria Paula Prata de Sousa que de bom agrado aceitou a minha sugestão, propondo a criação de uma aplicação web que permitisse aos alunos colocarem dúvidas sobre as aulas das suas Unidades Curriculares.

Para além do mais, esta aplicação recorre ao uso da linguagem C# e da *framework Asp.Net Core*, que se encontra num crescimento. Exemplos de alguns sites importantes que recorrem ao uso desta *framework* são a Microsoft, o Visual Studio, o StackOverflow, dentre outras, o que me motivou ainda mais para a realização desta aplicação.

1.3 Objectivos

O objectivo deste trabalho é criar uma aplicação que permita aos alunos colocar dúvidas aos professores em relação a uma UC. Para isso, no final de cada dia caso existam dúvidas colocadas a um professor, um email é enviado para o docente da cadeira a informar que tem questões em aberto. Quando o Professor responder às dúvidas postas pelos estudantes, esses mesmos serão notificados de que podem consultar as respostas no site. Se o professor selecionar apenas a visualização da resposta para o estudante que fez a questão e para um grupo restrito de alunos, os outros alunos apenas conseguem visualizar a resposta caso optem por seguir a dúvida posta.

Uma outra opção disponibilizada pela aplicação é a permissão de criar fóruns com questões em aberto.

1.4 Organização do Documento

De modo a reflectir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objectivos e a respectiva organização do documento.
2. O segundo capítulo – **Estado da Arte e Contextualização Tecnológica** – Expõe o contexto em que se insere a aplicação desenvolvida, apresenta algumas bibliotecas usadas e também algumas aplicações semelhantes.
3. O terceiro capítulo– **Engenharia de Software** – Apresenta os requisitos funcionais e não funcionais do sistema, descreve os casos de uso e a base de dados usada.
4. O quarto capítulo– **Implementação e Testes** – descreve detalhadamente a forma como as funcionalidades principais da aplicação web foram implementadas.
5. O quinto capítulo– **Conclusões e Trabalho Futuro** – descreve todas as conclusões que surgiram após a realização deste projeto, e algumas sugestões para um trabalho futuro.

Capítulo

2

Estado da Arte e Contextualização Tecnológica

2.1 Introdução

Neste capítulo, numa primeira introdução irá ser exemplificada a arquitetura utilizada pela *framework Asp .Net Core*, deste modo, vai mostrar a importância desta mesma para o desenvolvimento de uma aplicação web nos dias de hoje e ainda dentro da secção da contextualização tecnológica iremos mostrar as bibliotecas usadas que permitiram um melhor manuseamento da aplicação e as ferramentas/tecnologias que auxiliaram na construção do site. Para além do mais, serão descritas algumas aplicações semelhantes.

2.2 Contextualização Tecnológica

O *ASPNet Development* é uma plataforma para a construção de aplicativos web empresariais robustos que respeitam todos os serviços necessários. Embora existam mais opções para o desenvolvimento de aplicações, esta é uma das escolhas mais utilizadas pelos desenvolvedores em virtude de ser uma plataforma que usa arquitectura Model View Controller (MVC), esta arquitetura permite-nos separar as *Views*, dos modelos e dos controladores, deste modo, a manutenção do código torna-se mais fácil de entender e de implementar. Os componentes MVC podem ser testados isoladamente e o desenvolvimento dos vários componentes podem ser executados paralelamente. Esta é uma plataforma que pode ser usada em qualquer sistema operativo, o que nos permite ganhar flexibilidade e para além do mais, permite-nos o

uso de algumas bibliotecas já criadas que facilitam o manuseamento de certas funcionalidades.

Arquitetura *.Asp Net Core*

A arquitetura define a forma como os desenvolvedores e as bibliotecas irão interagir com a plataforma, desta forma, vamos aprofundar a arquitetura *.Asp Net Core* apresentada na imagem 2.1.

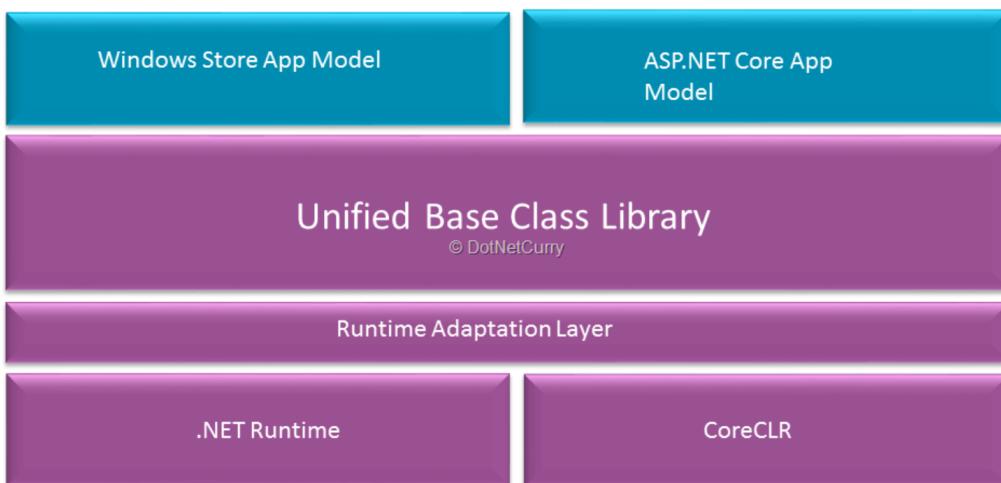


Figura 2.1: Arquitetura ASP.NET Core[2].

Antes de uma aplicação .NET Core ser executada, o código é convertido para uma linguagem intermediária, neste modo o *Common Language Runtime (CoreCLR)* fica responsável por converter este código em linguagem de máquina. O *.Net Runtime* também conhecido como *Managed Runtime* contém as bibliotecas nativas baseadas no Windows. Para além disto, permite a compilação *Ahead Of Time* o que melhora o desempenho dos aplicativos.

A *Unified Base Class Library* também conhecida como CoreFX, consiste nas classes básicas e fundamentais que formam o núcleo da plataforma .Net Core.

Para finalizar o *Windows Store App Model* é usado para o desenvolvimento de aplicativos Windows enquanto que o *Asp .Net Core App Model* é usado para o desenvolvimento web.

2.2.1 Bibliotecas Usadas

Tal como foi referenciado na secção 2.2 esta plataforma permite o uso de algumas bibliotecas já construídas que ajudam na implementação de certas funcionalidades.

A biblioteca ***Microsoft.AspNetCore.Identity*** é uma biblioteca que oferece suporte à função do registo e do *login*, assim como em termos de segurança. Esta biblioteca cria uma tabela chamada "*Users*" onde vai conter determinados campos (*PasswordHash* e *EmailConfirmed*) que permitem a segurança do utilizador como a integridade do site, esta tabela e os seus campos vão ser aprofundadas mais à frente na secção 3.4.

Outra utilidade importante da biblioteca é que permite-nos adicionar papéis/funções aos utilizadores, por exemplo um utilizador que seja um professor vai ter associado à sua conta o papel de professor, enquanto que um aluno vai-se associar ao papel de aluno. Se usarmos esta função com uma outra biblioteca denominada ***Microsoft.AspNetCore.Authorization*** podemos restringir o acesso às páginas que quisermos.

Por fim, a plataforma disponibiliza o acesso à biblioteca ***Microsoft.AspNetCore.Authentication*** esta biblioteca, contém os processos que permitem obter as credenciais dos utilizadores e usar essas credenciais para verificar a identidade do mesmo.

2.2.2 Ferramentas Usadas

Para a realização deste projeto, houve duas ferramentas importantes. A primeira a ser usada foi a plataforma *Azure*[3], esta é uma plataforma de *cloud* que permite compilar, executar e fazer a gestão de aplicações em várias *clouds*. Com o recurso a este site, foi possível criar um servidor de uma base de dados remota que corresse na aplicação.

A outra ferramenta importante que me auxiliou no design do site foi a aplicação *Nicepage*[4], esta ferramenta permite a construção de *websites* de forma gratuita, o que me permitiu aperfeiçoar o design das páginas inicialmente criadas.

2.3 Estado de Arte

O primeiro passo e um dos mais importantes para a execução deste projeto foi a pesquisa realizada com o intuito de encontrar aplicações semelhantes que auxiliaram na construção da minha aplicação, desta forma, sempre que surgiu um impasse ia complementar a documentação fornecida no enunciado do projeto com as aplicações já existentes.

2.3.1 Stackoverflow[1]

O *Stackoverflow* é um site de perguntas e respostas para programadores profissionais e iniciantes, ver imagem 2.2, ele apresenta perguntas e respostas em uma gama variada de tópicos. Segundo os seus criadores, ele foi criado para servir de alternativa aos sites de perguntas e respostas criados anteriormente.

O site serve como uma plataforma onde os utilizadores fazem perguntas e respondem a perguntas de outros utilizadores de uma forma gratuita. Os utilizadores conseguem desbloquear novos privilégios com o aumento na reputação, como a capacidade de votar, comentar e até mesmo editar as perguntas de outras pessoas, essa reputação aumenta com a participação activa no site, quer a fazer perguntas como a responder.

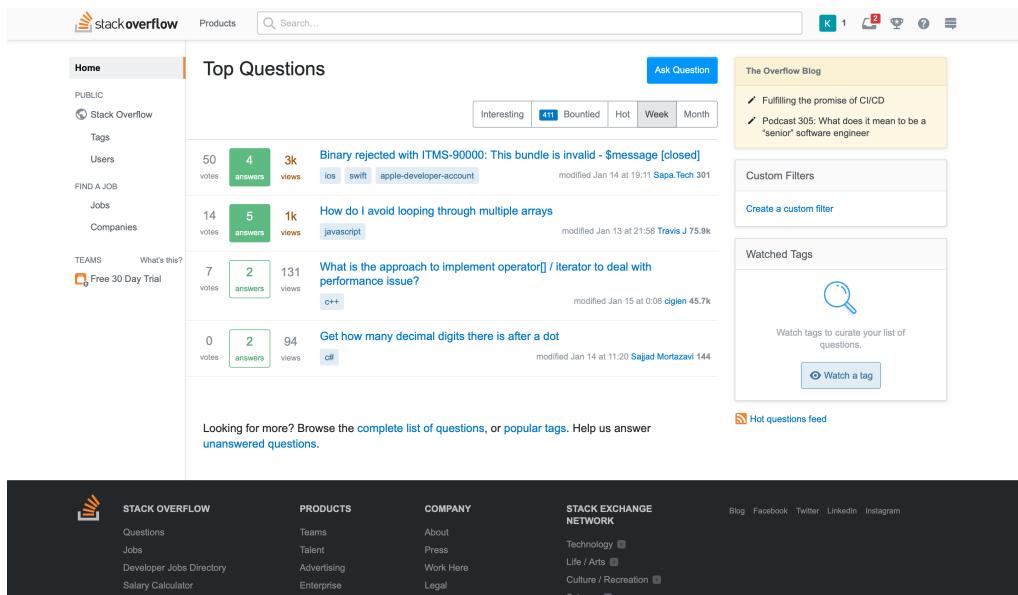


Figura 2.2: Página do Statckoverflow

2.3.2 Experts-Exchange

O site *Experts-Exchange* foi o primeiro site de perguntas e respostas a ser lançado para a área de programação, este foi lançado em Outubro de 1996. A primeira pergunta feita neste site foi "Case sensitive Win31 HTML Editor"[5] como podemos observar na imagem 2.3. Em 2001 a empresa faliu após uma mudança para *San Mateo*, tendo que mudar a sua gerência, os seus proprietários são *Austin Miller* e *Randy Redberg*.

O maior problema desta aplicação web e que causou uma grande polémica relaciona-se com o facto de apenas fornecer respostas a utilizadores

com uma assinatura paga. Ou seja, sempre que um utilizador deseja abrir uma pergunta este é reencaminhado para uma página onde é solicitada a compra anual ou mensal das funcionalidades do site, devido a este pormenor, os criadores do *Stackoverflow* decidiram iniciar o desenvolvimento da aplicação descrita na secção anterior (2.3.1).

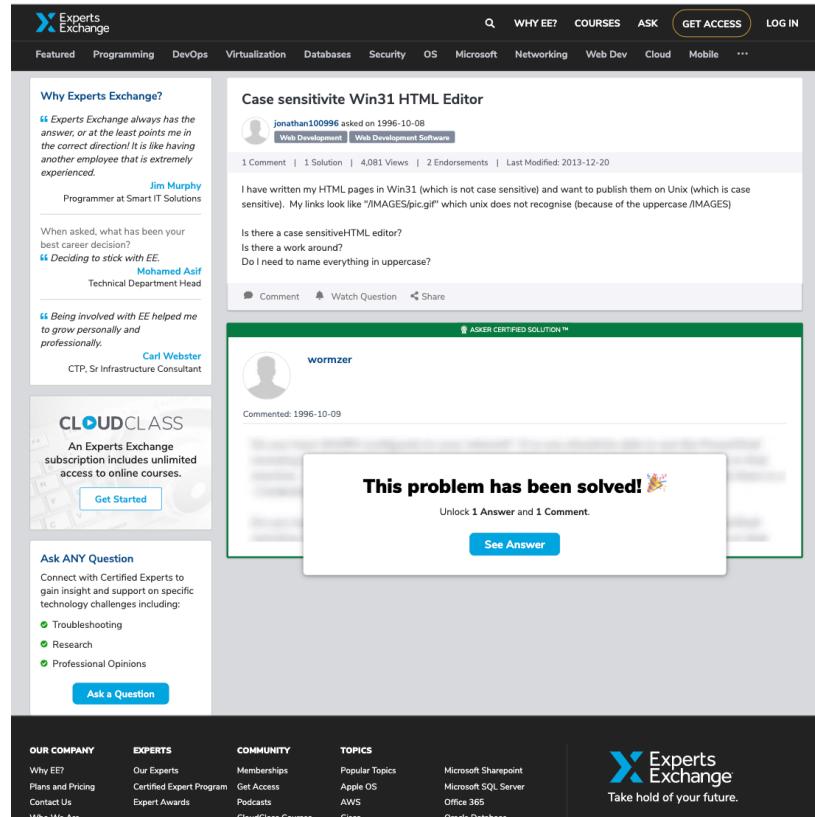


Figura 2.3: Case sensitive Win31 HTML Editor

2.4 Conclusões

Com a investigação concluída, consegui-me aperceber que não havia uma aplicação que permitisse aos alunos fazer questões que não estivessem relacionadas com a área da programação. De outro modo como podemos observar na aplicação do *Stackoverflow*, quem responde às perguntas tanto pode ser um aluno como um professor ou um engenheiro, o que nos leva a subentender que este é mais um fórum do que um site de perguntas e respostas. Porém a aplicação *Experts-Exchange* é um site onde só profissionais encarregues é que tem a permissão de responder a dúvidas. Deste modo, a aplicação

a ser desenvolvida vai ser uma aplicação que usa uma interligação entre as funcionalidades existentes nos dois sites e que seja gratuita para os alunos.

Capítulo

3

Engenharia de Software

3.1 Introdução

Ao longo deste capítulo, vai ser abordada a análise de requisitos (funcionalidades da aplicação), serão expostos os casos de uso de algumas páginas, deste modo, só irão ser abordadas as páginas mais importantes para fornecerem uma breve introdução de como a aplicação funciona.

E para finalizar, será exposta a base de dados com uma breve explicação de cada tabela tal como as conclusões finais obtidas desta secção.

Para uma melhor compreensão das funcionalidades da aplicação, vai ser explicado brevemente o objectivo da mesma. Nesta aplicação existem 3 tipos de utilizadores: admin, professor e aluno. A finalidade principal dos alunos é criar dúvidas/perguntas a uma determinada cadeira para que os professores mais tarde possam responder. Os professores diariamente recebem um email caso tenham dúvidas por responder, para além do mais, estes só podem responder a essas questões colocadas caso leccionem a cadeira em questão. Para finalizar, o administrador tem o poder de controlar os dados do servidor, ou seja, estes podem criar novos cursos, novas cadeiras, entre outras coisas. Para além do mais, caso o pretendam também podem apagar utilizadores do tipo aluno ou professor e também podem apagar ou editar cursos e cadeiras já existentes.

3.2 Análise de Requisitos

Nesta secção apresentam-se os requisitos funcionais que podemos observar na tabela 3.1, e que representam o que o software faz em termos de tarefas e serviços. Para além destes ainda existem os requisitos não funcionais que

se encontram presentes na tabela 3.2, estes estão relacionados com o uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas .

Requisitos Funcionais

Número	Descrição
RF-1	A aplicação permite o registo de novos utilizadores do tipo aluno ou professor.
RF-2	A aplicação permite a autenticação dos utilizadores.
RF-3	A aplicação permite alterar a palavra-passe caso um utilizador o deseje.
RF-4	A aplicação permite a um professor/aluno criar um fórum.
RF-5	A aplicação permite a um professor/aluno editar os seus fóruns.
RF-6	A aplicação permite a um professor/aluno apagar os seus fóruns.
RF-7	A aplicação permite a um professor/aluno fechar os seus fóruns.
RF-8	A aplicação permite a um professor/aluno visualizar as dúvidas existentes.
RF-9	A aplicação permite a um aluno criar uma dúvida.
RF-10	A aplicação permite a um aluno editar as suas dúvidas.
RF-11	A aplicação permite a um aluno apagar as suas dúvidas.
RF-12	A aplicação permite a um aluno seguir uma dúvida.
RF-13	A aplicação permite a um aluno parar de seguir uma dúvida.
RF-14	A aplicação permite a um professor criar um curso.
RF-15	A aplicação permite a um professor criar uma cadeira.
RF-16	A aplicação permite a um professor responder a uma dúvida.
RF-17	A aplicação permite a um professor selecionar se a resposta pode ser vista por todos os alunos ou por um grupo de utilizadores(quem criou a dúvida e quem a segue).
RF-18	A aplicação permite a um aluno/professor/administrador visualizar os cursos existentes.
RF-19	A aplicação permite a um aluno/professor/administrador visualizar as cadeiras existentes.
RF-20	A aplicação permite a um aluno/professor/administrador visualizar os fóruns existentes.
RF-21	A aplicação permite a um administrador apagar um curso.
RF-22	A aplicação permite a um administrador apagar uma cadeira.

RF-23	A aplicação permite a um administrador remover um docente de uma cadeira.
RF-24	A aplicação permite a um administrador remover um aluno de uma cadeira.
RF-25	A aplicação permite a um administrador apagar um fórum.
RF-26	A aplicação permite a um administrador editar um fórum.
RF-27	A aplicação permite a um administrador editar um curso.
RF-28	A aplicação permite a um administrador editar uma cadeira.
RF-29	A aplicação permite a um administrador fechar um fórum.
RF-30	A aplicação permite a um utilizador autenticado fazer o <i>logout</i> .

Tabela 3.1: Requisitos Funcionais

Requisitos Não Funcionais

Número	Descrição
RNF-1	A aplicação envia um email para confirmar o registo de um utilizador.
RNF-2	A aplicação envia um email quando um utilizador pretender mudar a sua palavra-passe.
RNF-3	A aplicação envia um email aos professores que tenham dúvidas por responder.
RNF-4	A aplicação envia um email ao criador de uma dúvida e aos seguidores dessa mesma dúvida a informar que um professor respondeu à pergunta.
RNF-5	A aplicação assegura a integridade dos dados de um utilizador.
RNF-6	A aplicação é compatível com dispositivos de diferentes resoluções.
RNF-7	A aplicação usa filtros de autorização.
RNF-8	A aplicação já contém um administrador registado na base de dados.

Tabela 3.2: Requisitos Não Funcionais

3.3 Casos de Uso

Nesta secção são demonstrados vários diagramas de casos de uso, que podem ser observados pelas seguintes figuras:

1. Caso de uso da página *login* (figura 3.1)
2. Caso de uso da página registo (figura 3.2)
3. Caso de uso do menu (figura 3.3)
4. Caso de uso da página visualizar dúvida (figura 3.4)

Login

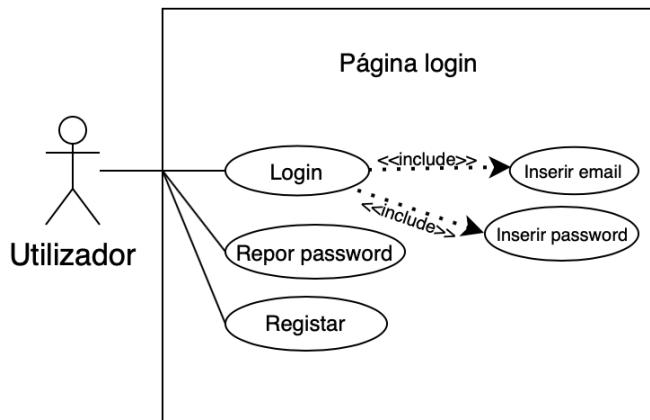


Figura 3.1: Caso de uso login.

Qualquer utilizador que pretenda fazer o *login* tem que estar previamente registado e com o email confirmado. Na página correspondente à figura 3.1 o utilizador pode fazer o *Login*, para isso, os dois campos (email e password) têm que estar preenchidos. Ou seja, não podem ser nulos e precisam de corresponder a um *Login* verdadeiro, senão irá aparecer uma mensagem a informar "Tentativa de *Login* inválida".

Caso algum dos campos não esteja preenchido, aparece a mensagem "Password é necessária" ou a mensagem "Email é necessário". Se o utilizador ainda não está registado, este pode escolher a opção "Registrar" que o vai reencaminhar para a página do registo, ou se este mesmo não se lembrar da password, pode escolher a opção "Repor password" que lhe vai permitir alterar a password através de um email que vai ser encaminhado.

Registrar

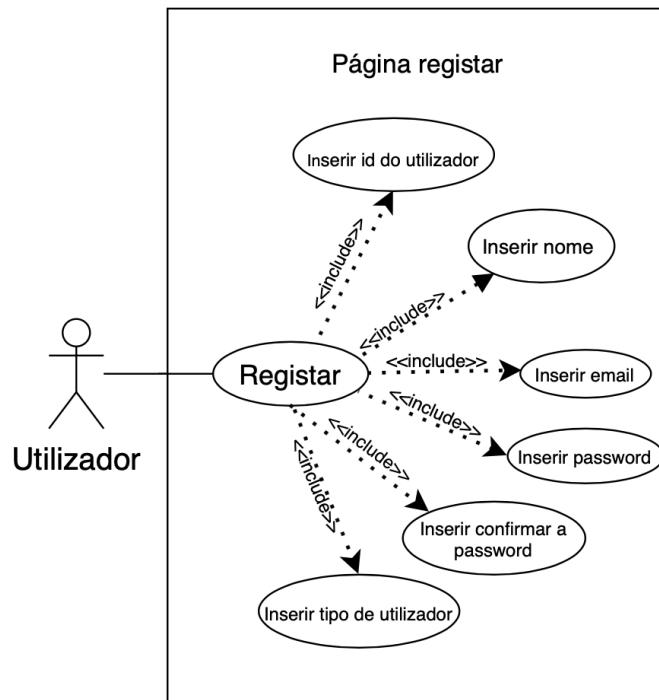


Figura 3.2: Caso de uso registo.

A ilustração 3.2 um utilizador não registado pode fazer o seu registo na plataforma, para isso precisa de preencher todos os campos, ou então vão aparecer mensagens semelhantes à da página *Login*. Se o utilizador for um aluno, no tipo de utilizador este precisa de seleccionar a opção "aluno", se for um professor o utilizador tem que escolher a opção "Professor". Deste modo, o administrador tem de verificar ocasionalmente se existem professores "falsos", ou seja, utilizadores do tipo Professor que não são docentes da UBI.

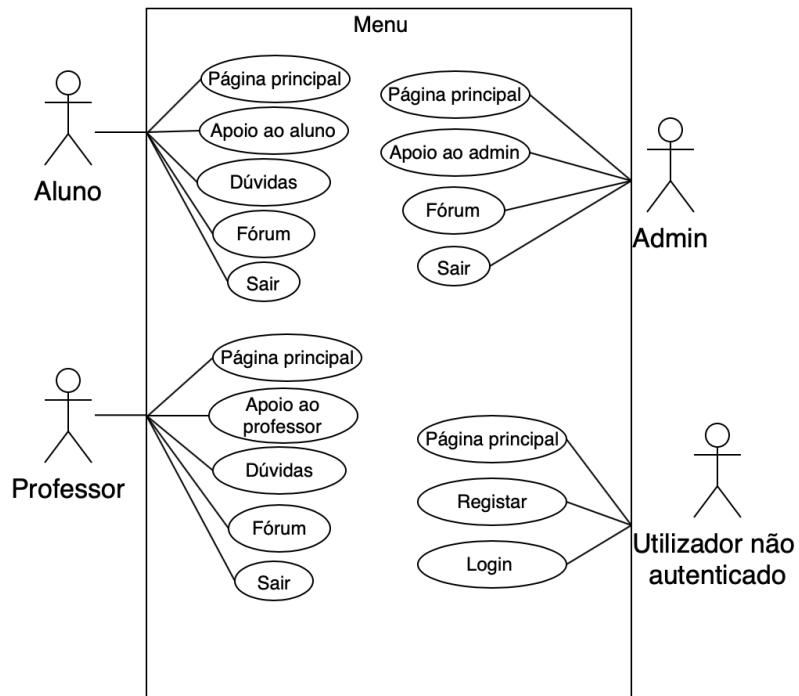
Menu

Figura 3.3: Caso de uso menu.

Para evitarmos o acesso a determinadas páginas, como podemos observar na figura 3.3 o menu só permite visualizar as opções que não vão restringir o acesso ao utilizador. Por exemplo, caso um aluno digitasse um link correspondente a uma página de um professor, este irá ser reencaminhado para uma página informativa onde notifica o utilizador que não contém as permissões necessárias para aceder à página em questão.

Visualizar Dúvida

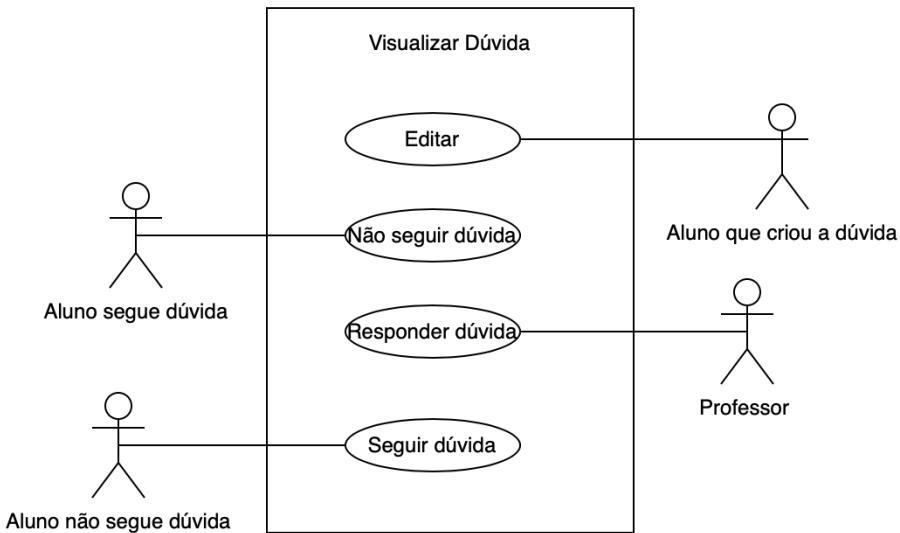


Figura 3.4: Caso de uso visualizar dúvida.

Consoante o utilizador, na página visualizar (imagem 3.4) dúvida e na página dúvidas podem aparecer diversas opções, se for o aluno que criou a questão vai-lhe aparecer a opção "Editar", caso seja um aluno que já siga a questão vai aparecer um coração branco preenchido a informar que apoia a mesma, ou então se estiver na página de visualizar apenas a questão seleccionada aparece um fixe preenchido, porém se não seguir a questão vai aparecer um coração ou um fixe sem estarem preenchidos a informar que o aluno não segue a questão e se quiser segui-la necessita de carregar em cima dos ícones, passando estes a estarem preenchidos. Porém se for um professor vai aparecer um ícone de um lápis que corresponde ao ícone editar/responder, deste modo, o professor sabe que pode responder à dúvida onde esse mesmo ícone está presente.

3.4 Base de Dados

A figura 3.5 demonstra a base de dados utilizada que permite o armazenamento dos dados usados na aplicação web.

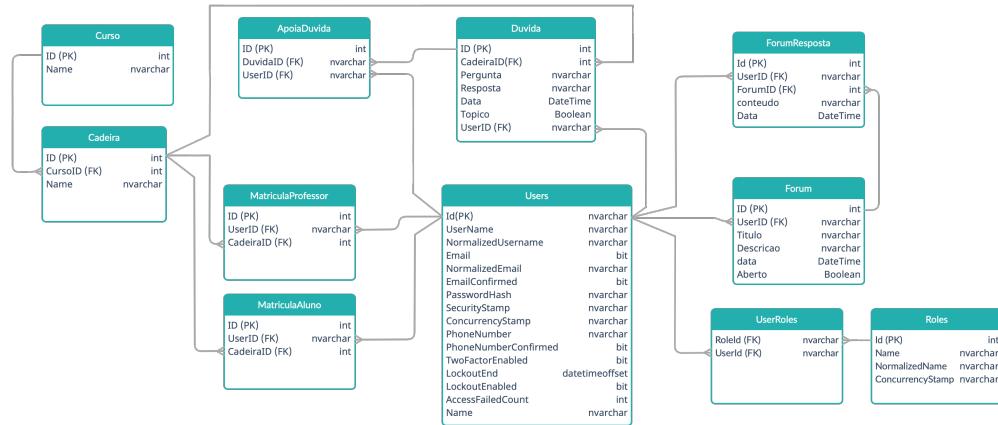


Figura 3.5: Base de dados.

Como foi referido na secção 2.2.1 a biblioteca *Microsoft.AspNetCore.Identity* ofereceu suporte às funcionalidades do registo e do *login*, de tal forma, esta biblioteca criou automaticamente as tabelas "Users", "UserRoles" e "Roles".

Com a criação destas tabelas, conseguimos garantir aos utilizadores que os seus dados são guardados em segurança, ou seja, a *password* não é gravada na base de dados e em vez disso vamos guardar a *PasswordHash*. Este *Hash* é o resultado final do *HMAC-SHA256* com 10000 iterações, para além do mais o *Hash* guardado na base de dados vai conter a junção do número de iterações com o *salt* e com a chave usada.

Deste modo, para verificarmos um utilizador precisamos de dividir a variável *PasswordHash* em 3 blocos em que o primeiro bloco corresponde ao número de iterações, o segundo bloco corresponde ao *salt* (tamanho 128 bits) e o último bloco corresponde à chave (tamanho 256 bits). Caso o utilizador tenha inserido os dados correctos o programa vai usar a *password* inserida na página *login* e o *salt* que saiu do segundo bloco da *PasswordHash* para gerar uma nova chave. Se a chave extraída da *PasswordHash* corresponder àquela que acabamos de gerar, então verificamos que o utilizador inseriu os dados correctos.

A tabela "Roles" contém os tipos de utilizadores disponíveis na aplicação, assim, com estes tipos de usuários e com a tabela "UserRoles" vamos associar cada utilizador a um papel/função.

A tabela "MatriculaProfessor" permite associar um utilizador do tipo professor a uma cadeira. Assim sendo, o docente consegue saber as cadeiras que lecciona. Por contrapartida, a tabela "MatriculaAluno" desempenha a mesma função só que esta é direcionada para um utilizador do tipo aluno permitindo-lhe seguir/matricular uma determinada cadeira.

As tabelas "Forum" e "Duvida" vão ser as tabelas que permitem guardar as informações de todas as dúvidas e de todos os fóruns. Com a existência da tabela "ForumResposta", conseguimos associar as respostas de diversos utilizadores a um determinado fórum e assim sendo, a tabela vai precisar de conter o *id* do fórum tal como o *id* do utilizador que criou a resposta. Deste modo, quando um utilizador estiver a observar um fórum consegue observar as respostas associadas e os criadores das mesmas.

No que diz respeito à tabela "ApoiaDuvida" podemos concluir pelo nome, que esta permite a um aluno apoiar uma determinada dúvida e assim, quando um professor disponibilizar a resposta a uma dúvida mesmo que tenha optado pela visualização restrita da resposta o aluno poderá observar a mesma.

Por último, a tabela "Curso" vai conter os cursos que os professores e os administradores criaram permitindo desta forma, associar a criação de uma "Cadeira" a um determinado curso.

3.5 Conclusões

A Engenharia de Software é um dos pontos mais fulcrais de qualquer projeto nos dias de hoje. Esta área facilita a compressão do funcionamento do sistema através dos casos de uso e da análise de requisitos. Para além do mais permite-nos entender a estruturação da base de dados que permite a persistência de dados nos sistemas.

Capítulo

4

Implementação e Testes

4.1 Introdução

Ao longo deste capítulo vai ser descrito o servidor web usado e todas as funcionalidades implementadas desta aplicação. Para além do mais, vão ser apresentados todos os testes realizados juntamente com os seus resultados.

4.2 Servidor

O *Kestrel* é um servidor web HTTP baseado em Entrada/Saída assíncrona (permite que o processamento continue antes que a transmissão termine), sendo usado em aplicações ASP .NET CORE. Este servidor é baseado na biblioteca de código aberto chamada libuv[6] e está disponível no GitHub[7].

Para além do mais este servidor permite configurar o HTTPS garantindo desta forma uma maior segurança do site. Para isso foi necessário criar um certificado digital X.509 auto-assinado. Porém como foi estudado na UC SI para um site ser realmente seguro seria preciso usar um certificado assinado por uma autoridade certificadora.

Como demonstração desta configuração, vai ser exemplificado o que foi feito para o servidor permitir o uso do HTTPS. O passo(4) só é possível ser feito num MAC mas não é obrigatório. O objectivo deste é evitar que apareça uma janela no *browser* a dizer que o site não é seguro. Para os utilizadores que não têm um Mac, caso esta janela apareça basta pressionar na opção que afirma que confiamos no site:

1. Comando executado na pasta AsMinhasDuvidas/AsMinhasDuvidas/certificadoeChaves para gerar uma chave privada e uma solicitação da assinatura de um certificado, através da configuração presente no ficheiro "https.config" que se encontra na pasta referida (ver excerto de código 4.1).

```
$ openssl req -config https.config -new -out csr.pem
```

Excerto de Código 4.1: Criação da chave privada.

2. Comando usado para criar um certificado auto-assinado (ver excerto de código 4.2).

```
$ openssl x509 -req -days 365 -extfile https.config -extensions v3_req -in csr.pem -signkey key.pem -out https.crt
```

Excerto de Código 4.2: Criação de um certificado auto-assinado.

3. Comando que permitiu gerar um ficheiro pfx, ficheiro que agrupa uma chave privada com o seu certificado, (ver excerto de código 4.3).

```
$ openssl pkcs12 -export -out https.pfx -inkey key.pem -in https.crt -password pass:<<Password>> /* password nao mostrada por questões de segurança */
```

Excerto de Código 4.3: Criação de um ficheiro pfx.

4. Comando executado para confiar no certificado. Este é o passo que só é possível ser realizado por um utilizador que contenha um Mac (ver excerto de código 4.4).

```
$ sudo security add-trusted-cert -d -r trustRoot -k /Library/Keychains/System.keychain https.crt
```

Excerto de Código 4.4: Comando para confiar no certificado.

5. Na classe Program foi adicionado o código ".UseKestrel(options => options.ConfigureEndpoints());" e a função correspondente ao "ConfigureEndpoints()" que se encontra presente no GitHub[8], onde vai ser lido o certificado (ver excerto de código 4.5).

```
public static IHostBuilder CreateHostBuilder(string[] args) =>
    Host.CreateDefaultBuilder(args) .
        ConfigureWebHostDefaults(webBuilder =>
    {
        webBuilder.UseStartup<Startup>().UseKestrel(
            options => options.ConfigureEndpoints());
    });
}
```

Excerto de Código 4.5: Código que permite configurar o kestrel.

6. Na classe appsettings.json foram adicionadas as configurações do "Http-Server" (ver excerto de código 4.6).

```
...
    "HttpServer": {
        "Endpoints": {
            "Http": {
                "Host": "localhost",
                "Port": 8080,
                "Scheme": "http"
            },
            "Https": {
                "Host": "localhost",
                "Port": 44340,
                "Scheme": "https",
                "FilePath": "certificadoechaves/https.pfx" /* 
                    caminho para o ficheiro pfx criado no Exerto
                    de Codigo 4.3 */
            }
        }
    ...
}
```

Excerto de Código 4.6: Configuração do HttpServer.

7. Comando executado para armazenar a senha do certificado como uma chave do utilizador durante o desenvolvimento (ver excerto de código 4.7).

```
$ dotnet user-secrets set HttpServer:Endpoints:Https:
    Password <password> /* password nao mostrada por
    questões de segurança */
```

Excerto de Código 4.7: Comando que guarda a string "HttpServer:Endpoints:Https:Password".

8. Alterações feitas ao ficheiro launchSettings.json para a aplicação usar o endereço HTTPS definido (ver excerto de código 4.8).

```
...  
  
"AsMinhasDuvidas": {  
    "commandName": "Project",  
    "launchBrowser": true,  
    "launchUrl": "https://localhost:44340",  
    "environmentVariables": {  
        "ASPNETCORE_ENVIRONMENT": "Development"  
    },  
    "applicationUrl": "http://localhost:8080/"  
}  
}  
}
```

Excerto de Código 4.8: Configuração do endereço HTTPS da aplicação.

9. Excerto de código retirado do GitHub [9] que permite o redirecionamento do site de HTTP para HTTPS (ver excerto de código 4.9).

```
...  
  
int? httpsPort = null;  
var httpsSection = Configuration.GetSection("HttpServer:Endpoints:Https");  
if (httpsSection.Exists())  
{  
    var httpsEndpoint = new EndpointConfiguration();  
    httpsSection.Bind(httpsEndpoint);  
    httpsPort = httpsEndpoint.Port;  
  
}  
var statusCode = env.IsDevelopment() ? StatusCodes.  
    Status302Found : StatusCodes.  
    Status301MovedPermanently;  
app.UseRewriter(new RewriteOptions().  
    AddRedirectToHttps(statusCode, httpsPort));  
  
...
```

Excerto de Código 4.9: Comando que permite o armazenamento da senha do certificado de forma segura.

Outra funcionalidade importante foi a implementação de *scheduled tasks* em que, com isto o servidor consegue verificar se tem que enviar um email a um professor. Ou seja, caso existam dúvidas por responder referentes ao dia actual do servidor, este vai informar o professor.

Desta forma, foram implementadas as classes que estão presentes no GitHub[10], porém em vez de recorrer ao uso da classe "HelloWorldJob", foi criada a classe "EmailJob" que permite verificar se o servidor precisa de enviar o email ou não.

Com as classes implementadas, tal como na implementação do HTTPS, só faltou adicionar na função "ConfigureServices" da classe "Startup.cs" os serviços que permitem a utilização das classes criadas. Deste modo, como conseguimos observar no excerto de código 4.10, este serviço vai correr todos os dias às 23:30.

```
...
services.AddSingleton<EmailJob>();
services.AddSingleton(new JobScheduler(
    jobType: typeof(EmailJob),
    cronExpression: "0 30 23 ? * *")); /*corre todos os dias s
                                         23:30, 0-segundos 33-minutos 23-horas ?-dia do mes(nao
                                         definido) *-todos os meses *-todas as semanas*/
services.AddHostedService<QuartzHostedService>();

...
```

Excerto de Código 4.10: Implementação do email JobScheduler.

4.3 Aplicação

Uma das aplicabilidades mais importantes deste projeto foi adaptar as páginas a vários tipos de dispositivos. De tal modo, algumas imagens que demonstram as páginas vão estar divididas em 4 imagens, em 2 imagens ou então numa. O objectivo da figura4.1 é representar os quatro tipos de dispositivos: smartphone, ipad, ipad pro e computador.

Com isto, esta aplicação consegue ganhar um maior número de utilizadores devido ao facto de não recorrem apenas ao uso dos computadores.

Por outro lado, quando aparecer uma imagem com duas secções, a secção de cima representa o ipad, o ipad pro e o computador. Enquanto que a de baixo representa o smartphone como podemos observar na figura 4.7.

Para finalizar, se for demonstrada apenas uma figura quer dizer que as páginas responsivas são muito semelhantes.

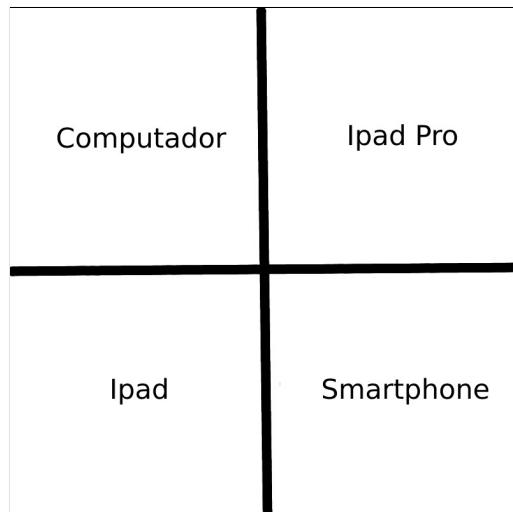


Figura 4.1: Demonstração das páginas responsivas.

4.3.1 Página Inicial

Relativamente ao *layout* das páginas o *header* e o *footer* como podemos observar na figura 4.2 pertencem ao *layout* relativo. Deste modo, estes vão aparecer em quase todas as páginas. As janelas que não recorrem ao uso deste *layout* só se vão diferenciar em relação ao menu que se encontra do lado esquerdo do *header*.

Quanto às funcionalidades podemos observar que só encontramos dois botões, o do "Login" envia o utilizador para a página do *login* e o botão "Registrar" permite um utilizador não autenticado registar-se na aplicação.



Figura 4.2: Página Inicial.

4.3.2 Menu

À excepção do que foi referenciado na secção 4.3, a figura 4.3 não representa as páginas responsivas. Nesta imagem são representadas as diversas opções que o menu pode apresentar.

Um utilizador não autenticado para além do acesso às três páginas principais (*login*, registar e Início) ainda pode observar os fóruns existentes.

Para além do mais, como podemos observar na figura a secção onde aparece "Bem vindo professor@ubi.pt" diz respeito a um utilizador do tipo professor. Ou seja, as opções que aparecem são as opções que um professor pode fazer. Se compararmos estas selecções com as de um utilizador do tipo aluno conseguimos aperceber-nos que só vai variar o "Apoio ao Professor" para "Apoio ao Aluno".

Consequentemente um administrador tem menos opções dado o facto de que nunca vai criar uma questão nem vai responder às dúvidas existentes.



Figura 4.3: Página Inicial.

4.3.3 Registrar

O primeiro passo desta aplicação é a funcionalidade do registo. Sem ela um aluno ou um professor não poderá usufruir do uso da aplicação. Devido a isto, a imagem 4.4 demonstra os campos necessários que um utilizador não registado tem que preencher.

Estes campos tal como os restantes da aplicação estão previamente definidos para evitar erros. Espaços referentes a um ID vão ser *inputs* do tipo número, áreas referentes a emails são *inputs* do tipo email e campos relacionados com palavra-psses são do tipo *password*. Zonas referentes a datas são do género data e por último campos semelhantes ao tipo de utilizador como está mostrado na figura vão ser do género *select*. Sendo assim, as opções que vão estar disponíveis no *select* da página registrar vão ser um string com o nome aluno e outra com o nome professor.

A figura 4.4 ainda permite representar as diferentes páginas responsivas como foi abordado na secção 4.3.

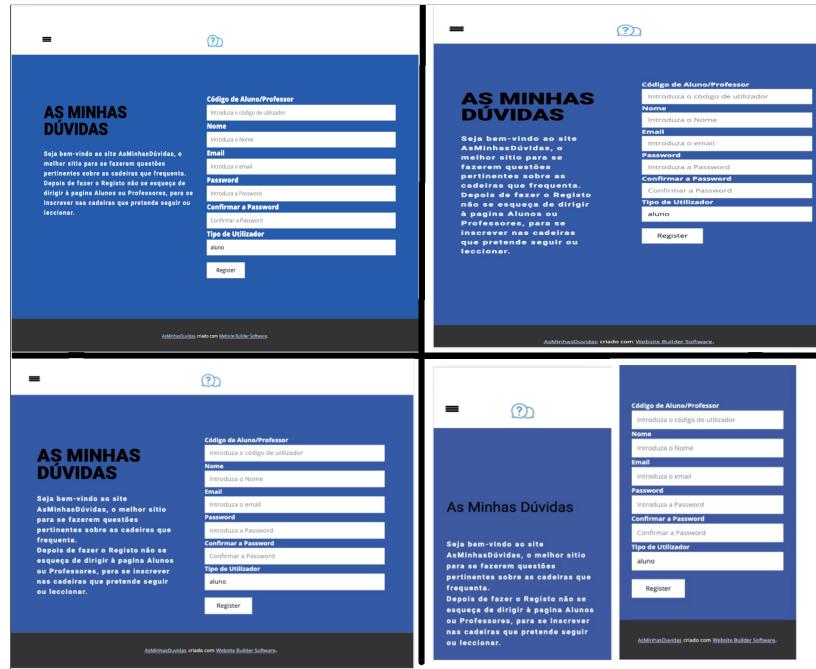


Figura 4.4: Página do registo.

4.3.4 Confirmar registo

Com a submissão correcta dos dados na página "Registar" a aplicação vai enviar o utilizador para uma janela informativa que pode ser observada na figura 4.5. Esta página já não utiliza o *layout* relativo como foi referenciado na secção 4.3.

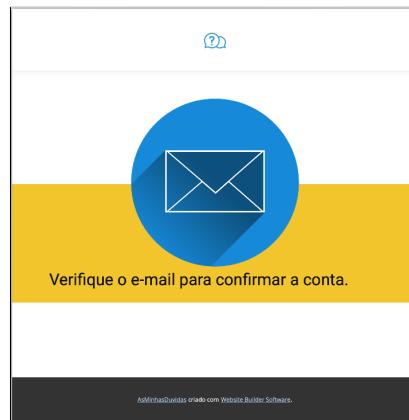


Figura 4.5: Página informativa do registo.

4.3.5 Login

Após a confirmação da conta o utilizador já pode efectuar o *login* caso o deseje. Porém enquanto esta não for confirmada ou se a submissão dos dados do login contiver algum campo errado, ou seja, qualquer acção que leve a um *login* inválido vai aparecer a mensagem "Tentativa de *login* inválida".

Tal como podemos observar na figura 4.6 o utilizador tanto pode fazer o *login* como o registo. Caso se tenha esquecido da *password* tem a hipótese de a repor como vai ser abordado nas duas secções seguintes.

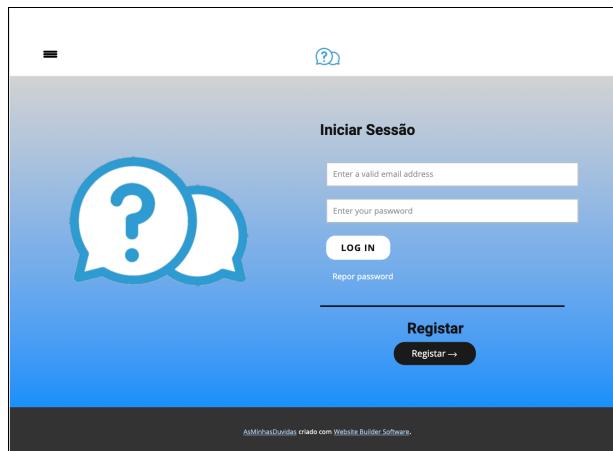


Figura 4.6: Página do login.

4.3.6 Repor password

Conforme a visualização da ilustração 4.7 a imagem com o boneco a pensar só vai aparecer nos dispositivos que respeitam as dimensões de um computador de um ipad ou de um ipad pro.

Caso o campo introduzido do email não seja válido (a conta não foi confirmada ou não existe na base de dados) vai ser mostrada uma mensagem informativa.

Por outro lado, se for bem sucedido, um email é enviado com o link que permite alterar a *password* da conta respectiva.



Figura 4.7: Página que permite repor a password.

4.3.7 Alteração de password

Na ilustração 4.8 é possível observar os *inputs* que permitem a alteração da palavra-passe. Para que isto aconteça o usuário precisa de introduzir a mesma chave nos campos correspondentes à password.

Caso estes campos sejam diferentes o servidor vai indicar ao utilizador que as chaves inseridas não correspondem.

Após a inserção correta dos dados o utilizador é enviado para outra página informativa como se pode observar na figura (imagem 4.9).

The screenshot shows a web page with a blue header and footer. In the center is a white rectangular form with a title 'REPOR PASSWORD'. It contains three input fields: 'Email', 'Password', and 'Confirmar Password', each with a horizontal line below it for text entry. Below these fields is a small rectangular button labeled 'REPOR'. At the bottom of the page, there is a small watermark-like text 'AdminWebOviedos criado com Website Builder Software'.

Figura 4.8: Página que permite alterar a password.

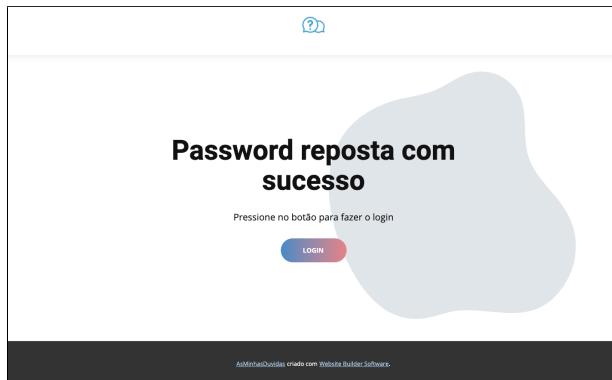


Figura 4.9: Página obtida após alteração da password.

4.3.8 Apoio ao aluno

Para evitar repetições as páginas do apoio ao aluno, do apoio ao professor e do apoio ao administrado vão ser demonstradas através de um utilizador que corresponde a um aluno.

A ilustração 4.10 corresponde ao *tab* das cadeiras onde um aluno pode procurar as cadeiras que existem na base de dados. Esta janela encontra-se de forma semelhante às páginas do apoio ao professor e do apoio ao administrador. A única diferença significativa nas páginas do Professor e do Admi-

nistrador é que vai aparecer um botão que permite criar uma cadeira ao lado da opção "Pesquisar".

Como se pode observar na ilustração o botão "anterior" encontra-se desactivado devido ao facto de corresponder à página inicial da tabela. Por sua vez, o botão "seguinte" encontra-se activado o que indica que existem mais cadeiras para visualizar.

A área "Cursos" não vai ser exemplificada devido às semelhanças que tem a esta página, a única distinção existente é que na tabela em vez de mostrar as cadeiras mostra os cursos e no apoio ao professor mostra o botão "Criar cadeira".

ID cadeira	Nome cadeira	ID curso	Nome curso
1	Programação 1	1	Engenharia Informática
2	Programação 2	1	Engenharia Informática
3	Programação Orientada aos Objectos	1	Engenharia Informática
4	Arquitetura de Computadores 1	1	Engenharia Informática

Figura 4.10: Tab da página apoio ao aluno que correspondente à área cadeira.

A subsecção "Frequenta" é a área que permite ao aluno apagar as cadeiras que segue (está matriculado). De tal forma, quando pretender remover uma matrícula, basta pesquisar a cadeira que pretende parar de seguir e de seguida precisa de pressionar o botão "Apagar" presente na tabela como está representado na figura 4.11. Na página correspondente a um utilizador do tipo professor em vez de o *tab* ter o nome "Frequenta" tem o nome "Lecciona".

Finalmente para um aluno/professor se poder associar a uma cadeira este precisa de seleccionar o curso a que pertence a cadeira e após isto deve escolher a cadeira que pretende seguir. Caso este introduza um id de uma UC em

que já esteja matriculado, uma mensagem é apresentada a informar o utilizador do mesmo.

The screenshot shows a web application interface titled "Apoio ao aluno". At the top, there are navigation links: "Cadeiras", "Cursos", "Matricular", and "Frequenta". Below these, a section titled "Secção Frequentada" contains a note: "Caso pretenda remover alguma cadeira das frequentadas, selecione na tabela a opção 'Apagar'". A table lists four courses:

ID cadeira	Nome cadeira	ID curso	
1	Programação 1	1	Apagar
2	Programação 2	1	Apagar
3	Programação Orientada aos Objectos	1	Apagar
4	Arquitetura de Computadores 1	1	Apagar

At the bottom of the table are two buttons: "Anterior" and "Seguinte".

Figura 4.11: Tab da página apoio ao aluno que permite parar de seguir uma cadeira.

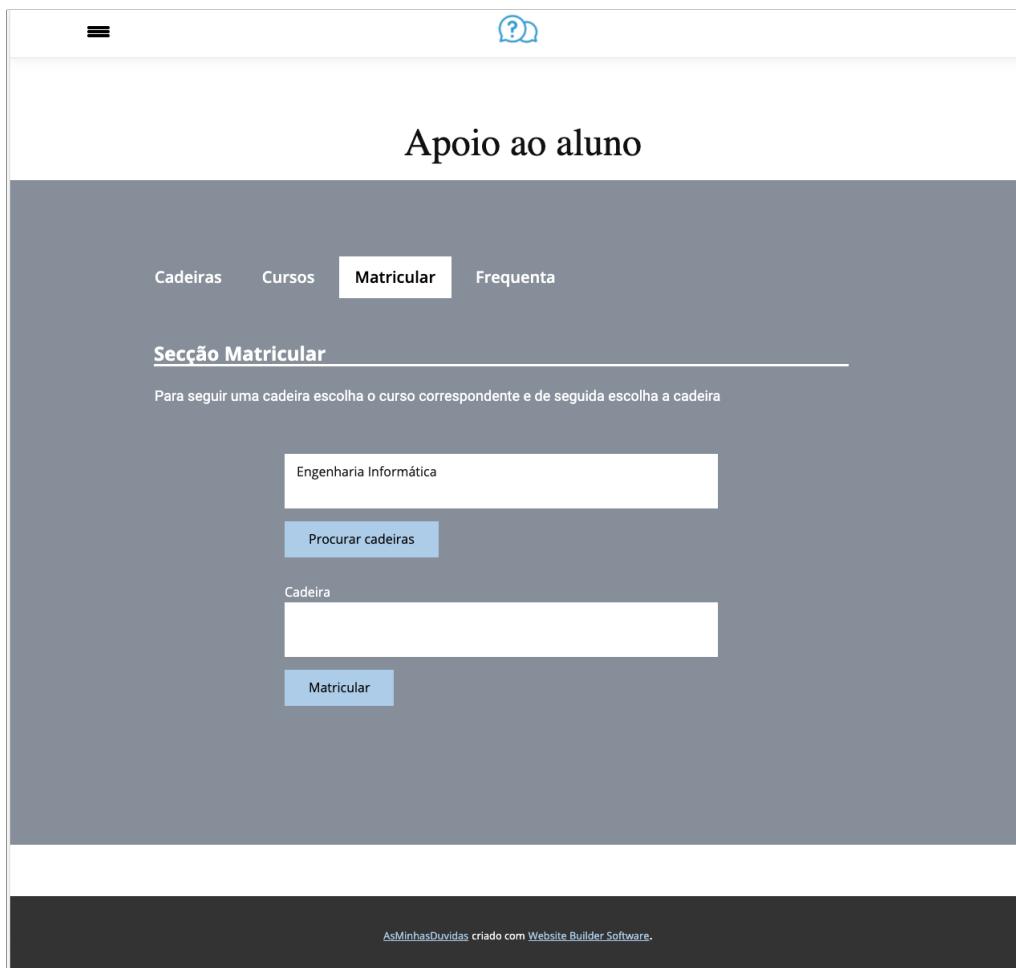


Figura 4.12: Tab da página apoio ao aluno que permite seguir uma cadeira.

4.3.9 Criar cadeira e curso

Tanto um professor como um administrador tem a capacidade para aceder às páginas que permitem criar cadeiras e cursos.

Como foi apresentado na secção da base de dados 3.4 para se criar uma cadeira é preciso referenciar o curso correspondente. Caso o curso ainda não exista o utilizador vai ter que o criar antes de decidir optar pela criação da cadeira. Na ilustração 4.13 aparecem os campos necessários para a criação de um curso, enquanto que na figura 4.14 vemos os campos pedidos para a criação de uma cadeira.

Figura 4.13: Página que permite criar um curso.

Figura 4.14: Página que permite criar uma cadeira.

4.3.10 Dúvidas

Após a inscrição nas cadeiras um aluno já está pronto para criar uma dúvida quando o pretender. Para isso, tem de dirigir-se à página "Dúvidas" do menu e em seguida deve carregar no botão "Criar Dúvida" como pode ser observado na figura 4.15. Devido a isto, esta opção não aparece aos docentes inscritos no site visto que nunca vão criar uma pergunta.

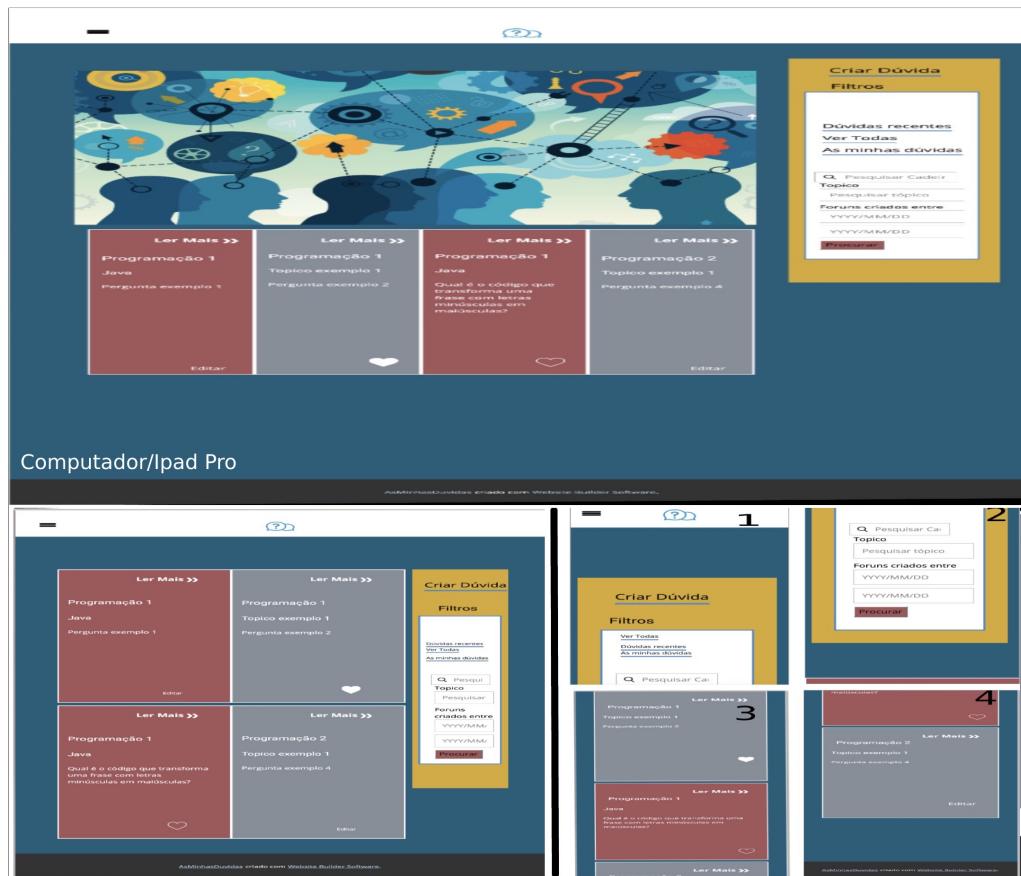


Figura 4.15: Página que lista as dúvidas existentes.

Na janela "Criar Dúvida"(ver figura 4.16) o aluno vai precisar de inserir uma *string* até 50 caracteres no espaço correspondente ao Tópico. Após a inserção do primeiro campo vai ser necessário escrever a dúvida e vai ser preciso seleccionar a cadeira relacionada com a dúvida.

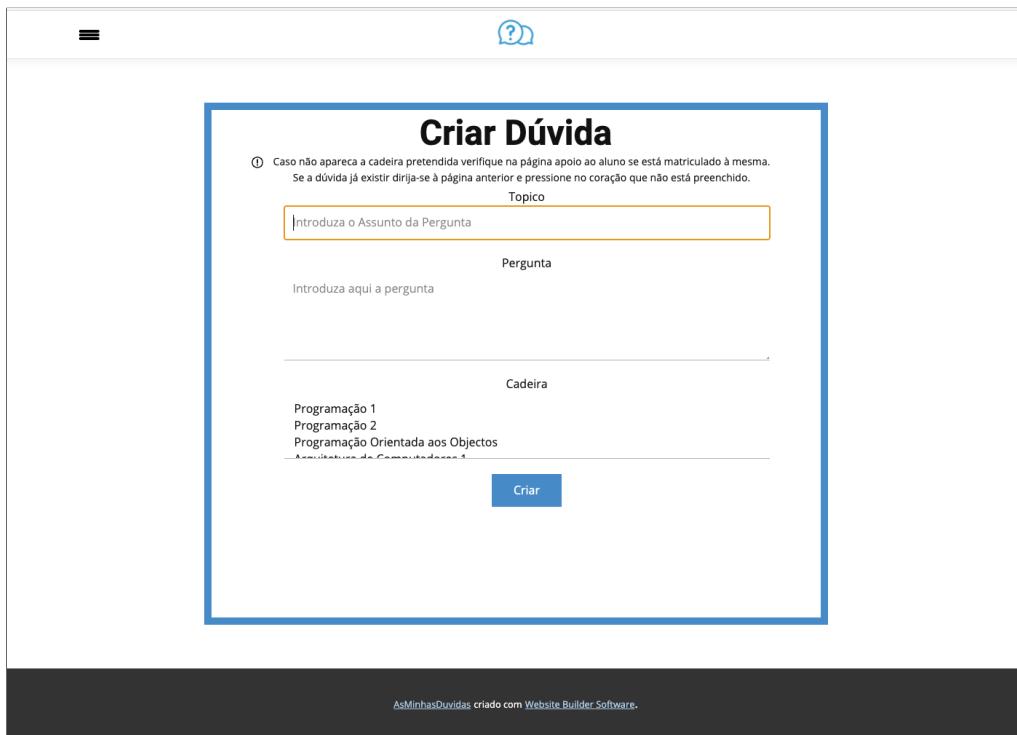


Figura 4.16: Criar dúvida.

A página informa o aluno para se dirigir à página anterior (figura 4.15) caso pretenda seguir alguma dúvida já criada. Para isso precisa pressionar o coração não preenchido (indica que o utilizador não segue a pergunta). Se aparecer um coração preenchido é porque o aluno já apoia a dúvida.

Porém ainda existe a possibilidade de aparecer um botão com a frase "editar", esta hipótese só aparece ao criador da dúvida.

No filtro de pesquisa que se encontra sobre a divisão a amarelo um aluno pode visualizar três botões. A primeira alternativa permite visualizar as dúvidas pela ordem decrescente da sua criação, ou seja, as dúvidas mais recentes aparecem primeiro. Em baixo do botão referido anteriormente encontramos a opção "Ver Todas" que permite remover os filtros de pesquisa. Estes filtros incluem a pesquisa do nome de uma cadeira, o tópico de uma dúvida e um intervalo de datas. Por último, o botão apresentado no fim permite a um aluno visualizar as dúvidas criadas por ele e as dúvidas que segue. Assim deste modo, facilita a procura das respostas às dúvidas que apoiaram ou que criaram.

Para embelezar esta janela foi limitada a visualização das questões, ou seja, a variável pergunta correspondente à tabela "Duvida" não tem nenhum

limite de caracteres. Se uma pergunta contiver mais de cem caracteres esta vai ser "alterada" apenas na página 4.15 mostrando assim os cem primeiro caracteres com a presença de reticências no final da frase. Se um aluno/professor quiser ver a dúvida completa tem de carregar no botão "Ler Mais »" o que vai fazer com que este seja enviado para a página "Visualizar Dúvida".

4.3.11 Visualizar Dúvida

Na ilustração 4.17 podemos observar os campos que pertencem à tabela "Dúvida". Relativamente à resposta se um professor ainda não disponibilizou a mesma vai aparecer uma mensagem no campo pertencente a informar que a mesma ainda não está disponível. Quando um docente fornecer uma resposta na página relativa, ver imagem 4.18, o docente tem a opção de escolher se a mesma vai estar disposta para todos os alunos ou só para um grupo restrito de alunos (criador da dúvida e quem segue a dúvida). Se a resposta só estiver para um grupo restrito de alunos na explicação da mesma aparece uma mensagem a informar que o utilizador não tem permissão para visualizar a mesma.

O botão eliminar que pode ser observado na primeira imagem só pode ser visualizado pelo criador da mesma dúvida, a função deste é eliminar a dúvida da base de dados. Enquanto que o lápis permite ao criador da questão alterar a pergunta ou então no caso de um professor envia-o para a página que permite responder à dúvida. No local do lápis podem existir outros botões que podemos observar na figura 4.17. Estes tem a mesma função que os botões da página anterior (onde estão listadas as dúvidas), ou seja, o *like* preenchido informa que segue a dúvida, o *like* sem estar preenchido informa que não segue a mesma e estes só podem ser visualizados por alunos que não criaram a dúvida.

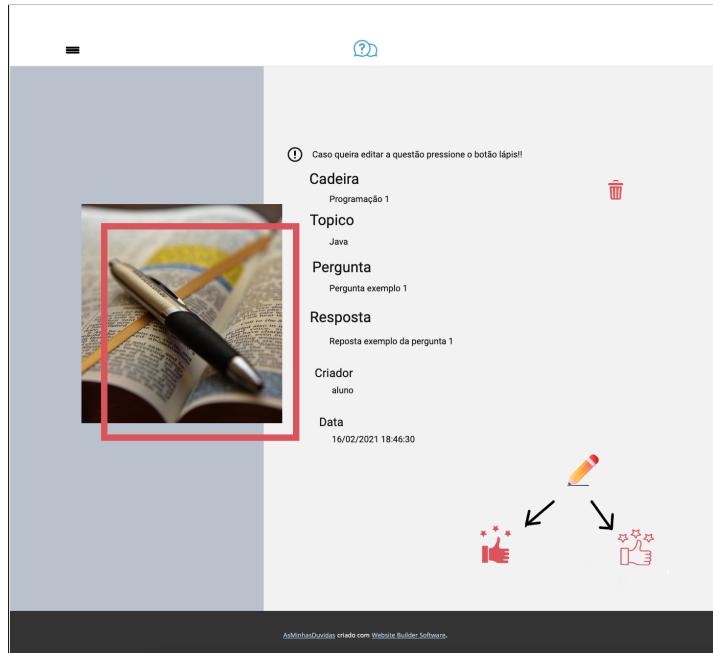


Figura 4.17: Visualizar dúvida.

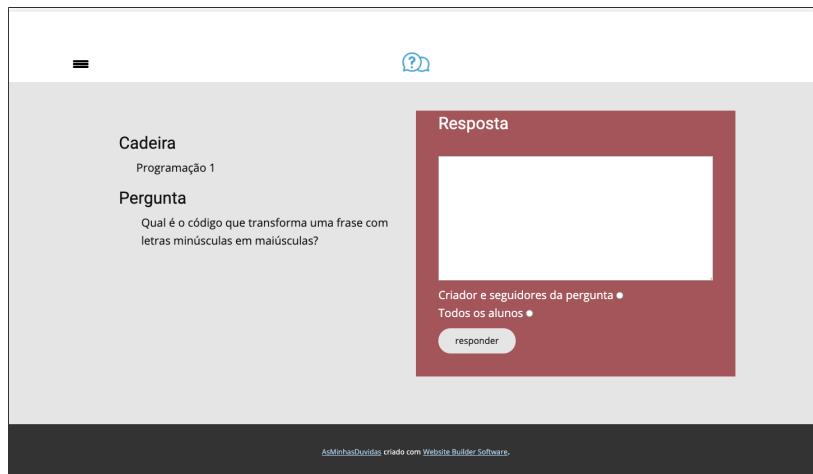


Figura 4.18: Resposta a uma dúvida.

4.3.12 Fórum

A página fórum como foi dito na subsecção menu 4.3.2 pode ser vista por qualquer tipo de utilizadores. Deste modo, como pode ser testemunhado na

figura 4.19 a opção "Criar Forum" apenas pode ser visualizada pelos utilizadores autenticados. Os utilizadores que não estão autenticados apenas podem visualizar os fóruns e respectivos *posts*.

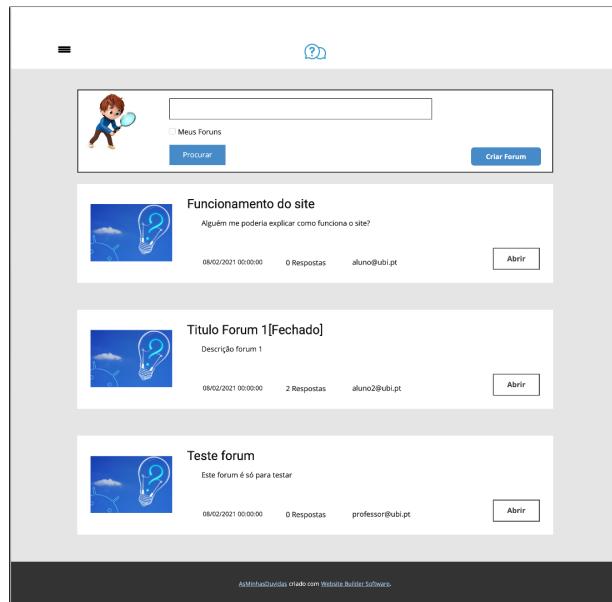


Figura 4.19: Visualização dos fóruns.

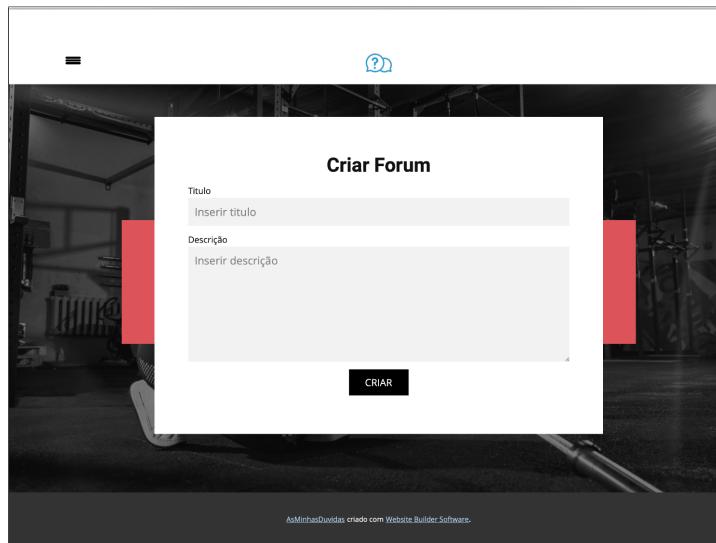


Figura 4.20: Criação de um fórum.

Para a criação de um fórum vão ser pedidos dois campos como podemos assistir na figura 4.20. O primeiro campo corresponde ao tópico do que vai ser falado ao longo dos *posts*, o segundo campo é a descrição da dúvida que um aluno/professor pretende obter resposta.

Para visualizar os posts/respostas de um determinado fórum, basta carregar a opção "abrir". Desta forma, vai abrir a janela correspondente (ver imagem 4.21) onde vão estar listados os posts existentes.

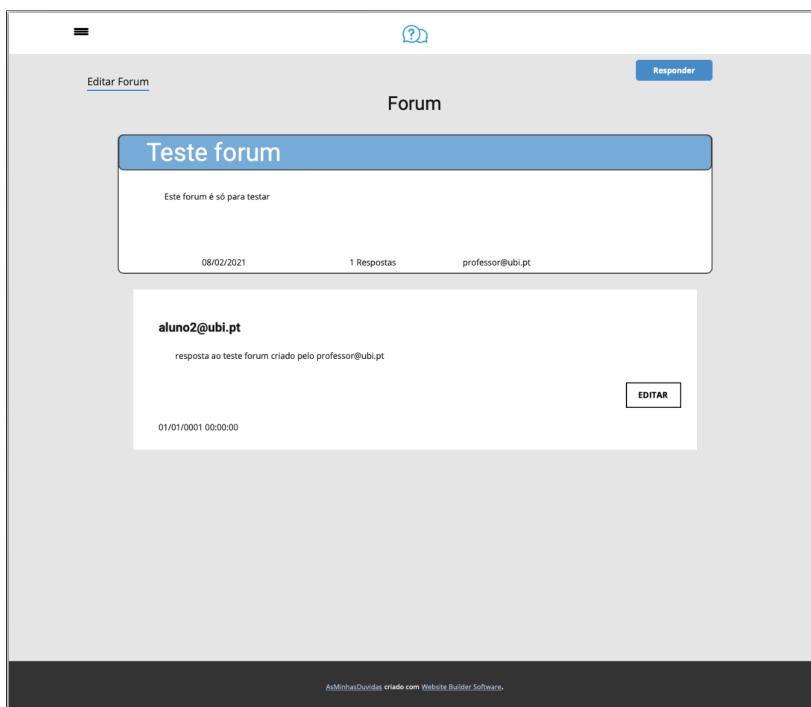


Figura 4.21: Visualização dos posts de um fórum.

Nesta janela, qualquer utilizador autenticado tem a opção "Responder" que permite criar um novo post (ver imagem 4.22). Para isto, o fórum tem que estar aberto, ou seja, o criador do fórum ou um administrador quando quiserem podem usar o botão que permite fechar a questão em causa. Essa opção vai abrir uma nova janela (ver figura 4.23) onde se pode alterar a descrição, pode-se alterar um título ou então ainda se pode apagar ou fechar o fórum em questão.

Quando um fórum estiver fechado as opções para editar e responder vão ser desabilitadas mantendo assim o fórum o mais simples possível. Evitando desta forma a criação de posts indesejáveis.

O forúm foi adicionada à aplicação web para permitir uma maior interacção entre os diversos utilizadores, facilitando a discussão de assuntos pertinentes.

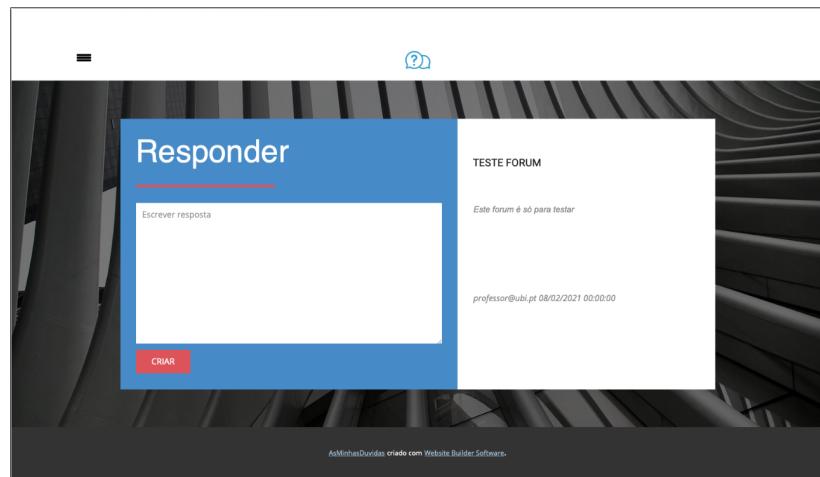


Figura 4.22: Criação de um post.

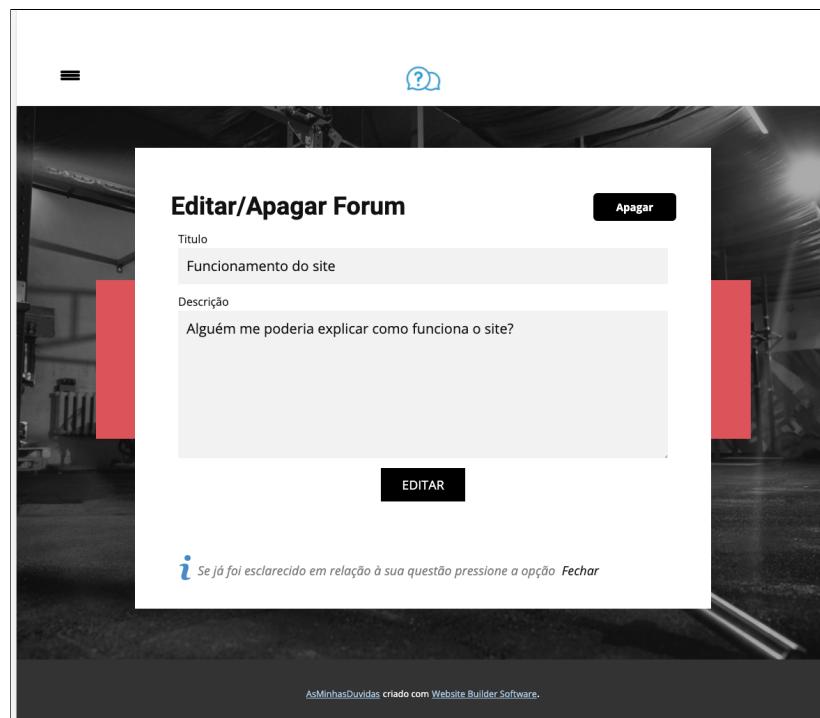


Figura 4.23: Editar fórum.

4.4 Segurança

No capítulo 2.2.1 foram ilustradas as diferentes bibliotecas usadas que garantem uma maior segurança do site. Desta forma, para demonstrar como estas bibliotecas podem ser usadas vão ser exemplificadas algumas anotações usadas ao longo da aplicação.

1. A anotação AllowAnonymous permite que um utilizador não autenticado possa visualizar a página (ver excerto de código 4.11).

```
namespace AsMinhasDuvidas.Pages.Foruns
{
    [AllowAnonymous]
    public class IndexModel : PageModel
    {
        ...
    } ...
}
```

Excerto de Código 4.11: Anotação AllowAnonymous.

2. Anotação Authorize garante que apenas utilizadores autenticados possam visualizar a página (ver excerto de código 4.12).

```
namespace AsMinhasDuvidas.Pages.Foruns
{
    [Authorize]
    public class CreateModel : PageModel
    {
        ...
    } ...
}
```

Excerto de Código 4.12: Anotação Authorize.

3. Anotação Authorize(Roles = "aluno") garante que a página ApoioAluno apenas possa ser visualizada por alunos (ver excerto de código 4.13).

```
namespace AsMinhasDuvidas.Pages.ApoioAluno
{
    [Authorize(Roles = "aluno")]
    public class CreateModel : PageModel
    {
        ...
    }
}
```

```
    } ...  
}
```

Exerto de Código 4.13: Anotação Authorize(Roles: "aluno").

4. Apenas professores podem visualizar a página (ver excerto de código 4.14).

```
namespace AsMinhasDúvidas.Pages.ApoioProfessor  
{  
    [Authorize(Roles = "professor")]  
    public class CreateModel : PageModel  
    {  
  
        ...  
  
    } ...  
}
```

Exerto de Código 4.14: Anotação Authorize(Roles: "professor").

5. Apenas administradores podem visualizar a página (ver excerto de código 4.15).

```
namespace AsMinhasDúvidas.Pages.ApoioAdmin  
{  
    [Authorize(Roles = "admin")]  
    public class CreateModel : PageModel  
    {  
  
        ...  
  
    } ...  
}
```

Exerto de Código 4.15: Anotação Authorize(Roles: "admin").

6. Comando que esconde a string "ConnectionString:DefaultConnection" na aplicação. Deste modo, só o criador da base de dados é que sabe a string que permite o acesso remoto à base de dados(ver excerto de código 4.17).

```
$dotnet user-secrets init  
$dotnet user-secrets set "ConnectionString:DefaultConnection" "  
    Server=tcp:asminhasduvidas.database.windows.net,1433;Database=  
    coreDB;User ID=rodrigosaraiva;Password=<passwordd>;Encrypt=true  
    ;Connection Timeout=30;" //password escondida por razoes de  
    seguranca
```

Excerto de Código 4.16: Comando que esconde a string "ConnectionStrings:DefaultConnection".

7. Comando que permite guardar a string "ConnectionStrings:emailpass" (password do email que é usado) na aplicação sem que esta esteja visível no ficheiro "appsettings.json" (ver excerto de código 4.18).

```
$dotnet user-secrets set "ConnectionStrings:emailpass" "<password>
" //nao mostrada por questões de segurança
```

Excerto de Código 4.17: Comando que esconde a password do email.

4.5 Testes

Para se testar as várias funcionalidades foram criados dados que vão ser embutidos na aplicação quando esta começa a correr, estes dados só são adicionados caso ainda não existam na base de dados. Com isto, foram testadas as funcionalidades de todos os botões da aplicação que permitiam apagar, pesquisar ou enviar os utilizadores para as páginas devidas. Os *inputs* dos formulários como foi abordado na subsecção 4.3.3 foram previamente definidos e os que são necessários apresentavam no código *html* o atributo *required*.

Por último foi testada a aplicação para verificar se os emails eram enviados para os utilizadores devidos.

4.6 Conclusões

Neste capítulo foram descritos os passos mais importantes para a configuração do servidor.

De seguida foram abordadas as funcionalidades da aplicação tal como os métodos de segurança embutidos que permitem ao servidor saber se o utilizador tem permissão para aceder aos recursos disponibilizados.

Por último foi explicado como foram feitos os testes na aplicação, sendo estes também abordados ao longo da explicação da aplicação (ex: mensagens informativas da submissão de dados).

Capítulo

5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Este projeto consistiu no desenvolvimento de uma aplicação web que permite a interacção dos alunos com os professores.

O desenvolvimento do projeto envolveu várias fases descritas ao longo deste relatório. Inicialmente comecei por fazer uma breve investigação de aplicações semelhantes, de seguida foi feito um estudo do estado da arte. Após isso foram identificadas as ferramentas e as tecnologias que iriam ser precisas para o desenvolvimento e para a sua posterior implementação.

Por último foram feitos vários testes à aplicação. A finalidade destes era verificarem se as funcionalidades impostas estavam a funcionar como esperado.

Para além do mais este trabalho realizado serviu para a consolidação de diversos conhecimentos leccionados ao longo da licenciatura em Engenharia Informática e também permitiu adquirir novos conhecimentos.

Esta aplicação no final revelou ser mais trabalhosa do que eu inicialmente esperava, o que me vai levar a saber gerir melhor o meu tempo para trabalhos futuros.

5.2 Trabalho Futuro

Embora todos os requisitos da aplicação web tenham sido implementados com sucesso, faltou uma peça fulcral do servidor. Por razões de tempo, não foi possível implementar o site num servidor remoto.

Para além do mais, poderiam ser acrescentadas mais algumas funcionalidades para aumentar a interactividade entre utilizadores. Algumas destas

funcionalidades sugeridas são:

1. Implementação de um chat online que permita a comunicação entre diferentes utilizadores.
2. Alteração da base de dados permitindo deste modo, aumentar a número de informações das cadeiras.
3. Efectuar a autenticação com o cartão de estudante da UBI.

Quando estas funcionalidades estivessem implementadas no sistema, a aplicação poderia ser proposta a diferentes agrupamentos escolares.

Bibliografia

- [1] Jeff Atwood and Joel Spolsky. Stackoverflow. [Online] <https://stackoverflow.com>. Último acesso a 17 de Janeiro de 2021.
- [2] Rahul Sahasrabuddhe. .NET Core: .NET of Future or Future of .NET? [Online] <https://www.dotnetcurry.com/dotnet/1308/dotnet-core-future-of-dotnet-framework>. Último acesso a 20 de Janeiro de 2021.
- [3] Microsoft. Microsoft Azure. [Online] <https://portal.azure.com>. Último acesso a 4 de Fevereiro de 2021.
- [4] Nicepage. Nicepage. [Online] <https://nicepage.com>. Último acesso a 4 de Fevereiro de 2021.
- [5] jonathan100996. Case sensitivite Win31 HTML Editor. [Online] <https://www.experts-exchange.com/questions/10000001/Case-sensitivite-Win31-HTML-Editor.html>. Último acesso a 17 de Janeiro de 2021.
- [6] Juan Pablo Canepa and Pat DeSantis. libuv. [Online] <https://github.com/libuv/libuv>. Último acesso a 6 de Fevereiro de 2021.
- [7] Nate McMaster. KestrelHttpServer. [Online] <https://github.com/aspnet/KestrelHttpServer>. Último acesso a 6 de Fevereiro de 2021.
- [8] Daniel Roth. KestrelHttps. [Online] <https://github.com/aspnet/samples/blob/master/samples/aspnetcore/security/KestrelHttps/KestrelServerOptionsExtensions.cs>. Último acesso a 7 de Fevereiro de 2021.
- [9] Daniel Roth. KestrelHttps. [Online] <https://github.com/aspnet/samples/blob/master/samples/aspnetcore/security/KestrelHttps/Startup.cs>. Último acesso a 7 de Fevereiro de 2021.
- [10] Andrew Lock. QuartzHostedService. [Online] <https://github.com/andrewlock/blog-examples/tree/master/QuartzHostedService/QuartzHostedService>. Último acesso a 9 de Fevereiro de 2021.